



HAL
open science

Clinica: an open source software platform for reproducible clinical neuroscience studies

Alexandre Routier, Ninon Burgos, Mauricio Díaz, Michael Bacci, Simona Bottani, Omar El-Rifai, Sabrina Fontanella, Pietro Gori, Jérémy Guillon, Alexis Guyot, et al.

► **To cite this version:**

Alexandre Routier, Ninon Burgos, Mauricio Díaz, Michael Bacci, Simona Bottani, et al.. Clinica: an open source software platform for reproducible clinical neuroscience studies. 2021. hal-02308126v2

HAL Id: hal-02308126

<https://inria.hal.science/hal-02308126v2>

Preprint submitted on 2 Apr 2021 (v2), last revised 16 Aug 2021 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Clinica: an open-source software platform for reproducible clinical neuroscience studies

Alexandre Routier^{1,2,3,4,5,6}, Ninon Burgos^{2,3,4,5,6,1}, Mauricio Díaz⁶, Michael Bacci^{1,2,3,4,5,6}, Simona Bottani^{1,2,3,4,5,6}, Omar El-Rifai^{1,2,3,4,5,6}, Sabrina Fontanella^{2,3,4,5,6,1}, Pietro Gori^{1,2,3,4,5,6}, Jérémy Guillon^{2,3,4,5,6,1}, Alexis Guyot^{2,3,4,5,6,1}, Ravi Hassanaly^{2,3,4,5,6,1}, Thomas Jacquemont^{2,3,4,5,6,1}, Pascal Lu^{2,3,4,5,6,1}, Arnaud Marcoux^{2,3,4,5,6,1}, Tristan Moreau^{2,3,4,5,6,8}, Jorge Samper-González^{2,3,4,5,6,1}, Marc Teichmann^{2,3,4,5,6,8,9}, Elina Thibeau-Sutre^{2,3,4,5,6,1}, Ghislain Vaillant^{2,3,4,5,6,1}, Junhao Wen^{2,3,4,5,6,1}, Adam Wild^{2,3,4,5,6,1}, Marie-Odile Habert^{10,11,12}, Stanley Durrleman^{1,2,3,4,5,6}, Olivier Colliot^{2,3,4,5,6,1,*}

¹ Inria, Aramis project-team, Paris, France

² Sorbonne Université, Paris, France

³ Institut du Cerveau – Paris Brain Institute - ICM, Paris, France

⁴ Inserm, Paris, France

⁵ CNRS, Paris, France

⁶ AP-HP, Hôpital de la Pitié-Salpêtrière, Paris, France

⁷ Inria Paris, SED, Paris, France

⁸ Institut du Cerveau – Paris Brain Institute - ICM, Paris, France

⁹ Department of Neurology, Institute for Memory and Alzheimer's Disease, Pitié-Salpêtrière Hospital, AP-HP, Paris, France

¹⁰ Sorbonne Université, CNRS, INSERM, Laboratoire d'Imagerie Biomédicale, LIB, Paris, France

¹¹ AP-HP, Hôpital Pitié-Salpêtrière, Médecine Nucléaire, Paris, France

¹² Centre d'Acquisition et Traitement des Images (CATI), www.cati-neuroimaging.com

* Correspondence:

Olivier Colliot, PhD – olivier.colliot@sorbonne-universite.fr

ICM – Paris Brain Institute

ARAMIS team Pitié-Salpêtrière Hospital

47-83, boulevard de l'Hôpital, 75651 Paris Cedex 13, France

Keywords: Neuroimaging, Software, Pipeline, Data processing, Machine learning, Multimodal data.

31 **Abstract**

32 We present Clinica (www.clinica.run), an open-source software platform designed to make clinical
33 neuroscience studies easier and more reproducible. Clinica aims for researchers to i) spend less time
34 on data management and processing, ii) perform reproducible evaluations of their methods, and iii)
35 easily share data and results within their institution and with external collaborators. The core of Clinica
36 is a set of automatic pipelines for processing and analysis of multimodal neuroimaging data (currently,
37 T1-weighted MRI, diffusion MRI and PET data), as well as tools for statistics, machine learning and
38 deep learning. It relies on the brain imaging data structure (BIDS) for the organization of raw
39 neuroimaging datasets and on established tools written by the community to build its pipelines. It also
40 provides converters of public neuroimaging datasets to BIDS (currently ADNI, AIBL, OASIS and
41 NIFD). Processed data include image-valued scalar fields (e.g. tissue probability maps), meshes,
42 surface-based scalar fields (e.g. cortical thickness maps) or scalar outputs (e.g. regional averages).
43 These data follow the Clinica Processed Structure (CAPS) format which shares the same philosophy
44 as BIDS. Consistent organization of raw and processed neuroimaging files facilitates the execution of
45 single pipelines and of sequences of pipelines, as well as the integration of processed data into statistics
46 or machine learning frameworks. The target audience of Clinica is neuroscientists or clinicians
47 conducting clinical neuroscience studies involving multimodal imaging, and researchers developing
48 advanced machine learning algorithms applied to neuroimaging data.

49 **1 Introduction**

50 Neuroimaging plays an important role in clinical neuroscience studies. While the meaning of clinical
51 neuroscience studies may vary, we use it to refer to studies involving human participants (i.e. patients
52 with neurological and psychiatric diseases, and control subjects) explored with multimodal data
53 (neuroimaging, clinical and cognitive evaluations, genetic data...) and most often involving
54 longitudinal follow-up. Carrying out such studies involves many data analysis steps including image
55 pre-processing, extraction of image-derived measurements and statistical analysis, thus requiring a
56 wide range of expertise. A similar situation is faced by researchers in machine learning for
57 neuroimaging: various steps are needed to extract features that are then fed to advanced learning
58 algorithms.

59 The first issue met when working on clinical studies concerns the organization of neuroimaging
60 datasets within or between institutions. The lack of a consistent structure makes arduous the sharing or
61 reuse of data. This is true for in-house, but also for publicly available neuroimaging datasets. Another
62 consequence of the lack of standard is the difficulty to apply automatic pipelines (e.g. extraction of
63 neuroimaging features, statistical analysis or machine learning) and to perform quality assurance. The
64 second issue faced by researchers processing data from clinical studies is related to the high number of
65 software packages, such as FreeSurfer¹ (Fischl, 2012), FMRIB Software Library (FSL)² (Jenkinson et
66 al., 2012) or Statistical Parametric Mapping³ (SPM) (Friston et al., 2007), that exist in the community.
67 Researchers have to understand the methodology behind each tool (e.g. segmentation, registration, etc.)
68 and master them from a programming perspective before being able to combine them and develop
69 image processing pipelines. Moreover, such “handicraft” approach makes it difficult to transmit tools
70 and knowledge, and to merge and share results of several studies due to the heterogeneous organization
71 of outputs. Finally, the difficulty to access or share both raw and processed neuroimaging data hinders
72 the reproducibility of neuroimaging studies (Poline et al., 2012).

73 Major progress has been made in the last years to ease neuroimaging studies. First, difficulties
74 related to the heterogeneity of image processing tools have been partly handled by the Nipype
75 (Neuroimaging in Python – Pipelines and Interfaces) software package⁴ (Gorgolewski et al., 2011).
76 Nipype is an open-source Python project that provides a uniform environment facilitating interaction
77 between neuroimaging software tools or algorithms, regardless of their programming language, within
78 a single workflow. Later, the issues related to the organization of the clinical and imaging data have
79 been tackled by the Brain Imaging Data Structure (BIDS) (Gorgolewski et al., 2016), a new standard
80 from the community for the community. The BIDS standard is based on a file hierarchy rather than on
81 a database management system, thus facilitating its deployment in any environment. Thanks to its clear
82 and simple way to describe neuroimaging and behavioral data, the BIDS standard has been easily
83 adopted by the neuroimaging community. Organizing a dataset following the BIDS hierarchy
84 simplifies the execution of neuroimaging software tools, resulting in the development of user-friendly
85 software. For instance, BIDS Apps (Gorgolewski et al., 2017) provides a set of pipelines for the
86 processing of neuroimaging data that follow a BIDS hierarchy. Currently, it mainly wraps
87 neuroimaging software packages from the community into a Docker image and is used via a command

¹ <http://surfer.nmr.mgh.harvard.edu/>

² <https://fsl.fmrib.ox.ac.uk/>

³ <http://www.fil.ion.ucl.ac.uk/spm/>

⁴ <https://nipype.readthedocs.io>

88 line interface. Moreover, the Nilearn⁵ (Abraham et al., 2014) package facilitates the application of
89 advanced machine learning approaches to neuroimaging data. To that purpose, it leverages the scikit-
90 learn library⁶ (Pedregosa et al., 2011) and provides tools for handling and visualizing different types
91 of neuroimaging data and building predictive models.

92 Nevertheless, carrying out a multimodal neuroimaging study remains challenging due to the
93 know-how necessary to grasp each modality and tool involved. While technical implementation has
94 been facilitated by Nipype, the development of a pipeline still requires substantial programming skills
95 and time to master both the neuroimaging software tools and Nipype. While the BIDS standard is being
96 adopted by the scientific community, not all public neuroimaging datasets provide a BIDS version of
97 their data. Besides, performing a single or multimodal neuroimaging study will also require
98 methodological expertise. For instance, a classification study of healthy subjects and patients with a
99 neurodegenerative disease using ¹⁸F-fluorodeoxyglucose positron emission tomography (FDG PET)
100 could involve notions of multimodal registration between FDG PET and T1-weighted magnetic
101 resonance imaging (MRI), tissue segmentation of T1-weighted (T1w) MRI, PET partial volume
102 correction, normalization into a standard space, and machine learning-based classification, as well as
103 know-how of the tools used to perform these steps. Moreover, the image processing steps need to be
104 chained from one to the other and the absence of data organization for processed neuroimages makes
105 data analysis more complex. Finally, the neuroimaging features generated by the pipelines need to be
106 correctly connected to statistical or machine learning frameworks.

107 Clinica (www.clinica.run) aims to make clinical research studies easier and pursues the
108 community effort of reproducibility. The core of Clinica is a set of automatic pipelines for processing
109 and analysis of multimodal neuroimaging data (currently, T1w MRI, diffusion MRI and PET data), as
110 well as tools for statistics, machine learning and deep learning. Clinica relies on tools written by the
111 scientific community and provides converters of public neuroimaging datasets to BIDS, processing
112 pipelines and organization for processed files, statistical analysis, and machine learning algorithms.

113 The target audience is mainly of two types. First, neuroscientists or clinicians conducting
114 clinical neuroscience studies involving multimodal imaging, typically not experts in image processing
115 for all of the involved imaging modalities or in statistical analysis. They will benefit from a unified set
116 of tools covering the complete set of steps involved in a study (from raw data to statistical analysis).
117 Second, researchers developing advanced machine learning algorithms, typically not experts in brain
118 image analysis. They will benefit from tools to convert public datasets into BIDS, fully automatic
119 feature extraction methods, and baseline classification algorithms to which they could compare their
120 results. Overall, we hope that Clinica will allow users to spend less time on data management and
121 processing, to perform reproducible evaluations of their methods, and to easily share data and results
122 within their institution and with external collaborators.

⁵ <https://nilearn.github.io>

⁶ <https://scikit-learn.org/>

123 2 Clinica overview

124 Clinica is an open-source software platform for reproducible clinical neuroimaging studies. It can take
125 as inputs different neuroimaging modalities, currently anatomical MRI, diffusion MRI and PET.
126 Clinica provides processing pipelines that involve the combination of different software packages. It
127 currently relies on FreeSurfer (Fischl, 2012), FSL (Jenkinson et al., 2012), SPM (Frackowiak et al.,
128 1997), Advanced Normalization Tools (ANTs)⁷ (Avants et al., 2014a), MRtrix3⁸ (Tournier et al.,
129 2012), and the PET Partial Volume Correction (PETPVC) toolbox⁹ (Thomas et al., 2016). The
130 pipelines are written using Nipype (Gorgolewski et al., 2011). Features extracted with the different
131 pipelines can be used as inputs to statistical analysis, which relies on SPM (Frackowiak et al., 1997)
132 and SurfStat¹⁰ (Worsley et al., 2009), or machine learning analysis, which relies on scikit-learn
133 (Pedregosa et al., 2011) and PyTorch (Paszke et al., 2019).

134 Input neuroimaging data are expected to follow the BIDS data structure (Gorgolewski et al.,
135 2016), as explained in section 3.2. Since this new standard has only recently been adopted by the
136 community, not all public neuroimaging datasets are yet proposed in BIDS format. To facilitate the
137 adoption of BIDS, Clinica curates several publicly available neuroimaging datasets and provides tools
138 to convert them into the BIDS format. Processed data are organized following the ClinicA Processed
139 Structure (CAPS) format, detailed in section 3.3, which shares the same philosophy as BIDS. Finally,
140 a set of tools is provided to handle input and output data generated by Clinica, thus facilitating data
141 management or connection to statistical or machine learning analysis.

142 The main functionalities of Clinica are described in the paper, but for further details the reader
143 can refer to the documentation available on the website¹¹. For each pipeline, the reader will find a
144 description of its functionalities, a list of the tools on which it relies, an example showing how to run
145 the pipeline, and a description of the outputs generated. The documentation of a pipeline can have
146 several levels of reading, which are respectively targeting people new or familiar with neuroimaging
147 and scientists working on pattern recognition and machine learning. User support is handled through a
148 forum¹² as well as using the issue tracker on GitHub¹³.

149

⁷ <https://stnava.github.io/ANTs/>

⁸ <http://mrtrix.org>

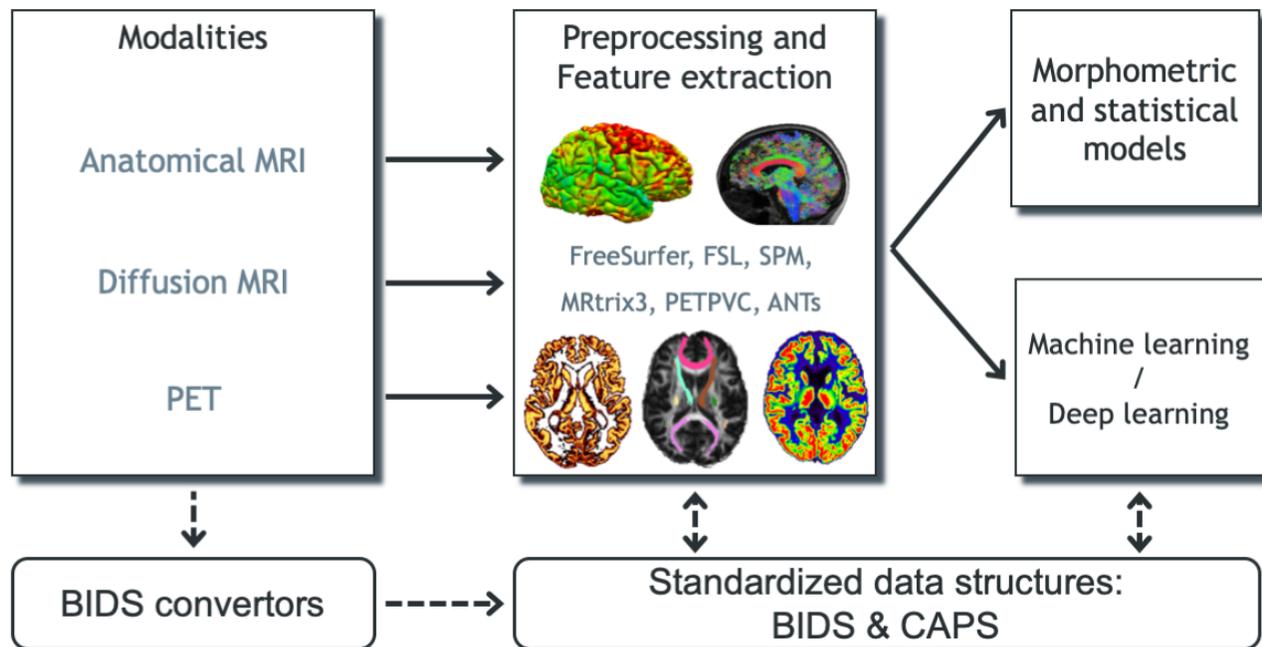
⁹ <https://github.com/UCL/PETPVC>

¹⁰ <http://www.math.mcgill.ca/keith/surfstat/>

¹¹ <http://www.clinica.run/doc>

¹² <https://groups.google.com/g/clinica-user>

¹³ <https://github.com/aramis-lab/clinica/issues>



150

151 **Figure 1.** Overview of Clinica’s functionalities. Clinica provides processing pipelines for MRI and
 152 PET images that involve the combination of different software packages, and whose outputs can be
 153 used for statistical or machine learning analysis. Clinica expects data to follow the Brain Imaging Data
 154 Structure (BIDS) and provides tools to convert public neuroimaging datasets into the BIDS format.
 155 Output data are stored using the ClinicaA Processed Structure (CAPS).

156

157 3 Clinica environment

158 3.1 Software architecture of Clinica

159 The core of Clinica is written in Python and mainly relies on the Nipype framework (Gorgolewski et
 160 al., 2011) to create pipelines. Python dependencies also include NumPy (van der Walt et al., 2011),
 161 NiBabel (Brett et al., 2019), Pandas (McKinney, 2010), NIPY, SciPy (Jones et al., 2001), scikit-learn
 162 (Pedregosa et al., 2011), scikit-image (van der Walt et al., 2011), nilearn (Abraham et al., 2014) and
 163 PyTorch (Paszke et al., 2019).

164 Clinica is provided to the end user in the form of a Python package distributed through Python
 165 Package Index (PyPI) and can simply be installed by typing `pip install clinica` through the
 166 terminal, within a virtual environment.

167 The main usage of Clinica is through the command line, which is facilitated by the support of
 168 autocompletion. The commands are gathered into four main categories. The first category of command
 169 line (`clinica run`) allows the user to run the different pipelines on neuroimaging datasets following
 170 a BIDS or CAPS hierarchy. The `clinica convert` category allows the conversion of publicly
 171 available neuroimaging datasets into a BIDS hierarchy. To help with data management, the `clinica`
 172 `iotools` category comprises a set of tools that allows the user to handle BIDS and CAPS datasets,
 173 including generating lists of subjects or merging all tabular data into a single TSV file for analysis with
 174 external statistical software packages. Finally, the last category (`clinica generate`) is dedicated to

175 developers and currently generates the skeleton for a new pipeline. Examples of command line can be
 176 found in Table 1.

177

178 **Table 1.** Categories of command line

Category of command line	Description	Example usage
run	Run pipelines on BIDS or CAPS datasets. The full list of pipelines is available in section Erreur! Source du renvoi introuvable..	<code>clinica run t1-linear bids_directory caps_directory</code>
convert	Convert public neuroimaging datasets into BIDS. The list of the public datasets can be found in section 5.1 .	<code>clinica convert adni-to-bids dataset_directory clinical_data_directory bids_directory</code>
iotools	Set of tools to handle BIDS and CAPS datasets. The list of the I/O tools can be found in section 5.2 .	<code>clinica iotools merge-tsv bids_directory --caps_directory caps_directory my_population.tsv</code>
generate	(For developers) Generate the skeleton source code for a new pipeline.	<code>clinica generate template "Modality Feature Extracted" pipeline_folder</code>

179 3.2 Input data with the BIDS standard

180 When dealing with multiple datasets, it is difficult to automate the execution of neuroimaging pipelines
 181 since their organization may vary from each other or even within each individual dataset. If we consider
 182 neuroimaging datasets involving many participants, the lack of a clear structure will necessitate a large
 183 amount of time to curate these databases and make them easily usable. Besides, large databases are
 184 often associated with database management systems, which involve additional technical and financial
 185 resources to be maintained.

186 BIDS (Gorgolewski et al., 2016) is a community standard enabling the storage of multiple
187 neuroimaging modalities and behavioral data. The BIDS standard provides a unified structure and
188 makes easier the development and distribution of code that uses neuroimaging datasets. Moreover, the
189 BIDS format is based on a file hierarchy rather than on a database management system, thus avoiding
190 the installation and maintenance of additional software. As a result, BIDS can be easily deployed in
191 any environment. The specification is intentionally based on simple file formats and folder structures
192 to reflect current laboratory practices, which makes it accessible to a wide range of scientists coming
193 from different backgrounds. People unfamiliar with the BIDS format can see an example of a BIDS
194 folder in Figure 3.

195 For these reasons, we also adopted this standard and Clinica expects that the input data are
196 BIDS-compliant for the execution of pipelines. Note that if a cross-sectional dataset (i.e. with no
197 `session` folder) is provided, Clinica will interactively propose to convert the cross-sectional dataset
198 into a longitudinal dataset with a unique session.

199

200 3.3 Input/Output data with the CAPS structure

201 Clinica has its own specifications for storing processed data, called CAPS (Clinica Processed
202 Structure). Of note, there exists an ongoing initiative called BIDS-derivatives that aims to provide a
203 BIDS standard for processed data. However, we wrote the CAPS specification before the start of the
204 BIDS-derivatives which explains why Clinica does not use the latter. Moreover, in their current state,
205 several outputs needed by Clinica are not covered or well adapted. In particular, the notion of group
206 does not exist yet. Nonetheless, we made humble contributions to BIDS-derivatives and we aim to
207 increasingly contribute. Ultimately, the two specifications will probably converge.

208 Processed data include image-valued scalar fields (e.g. segmentation labels, tissue maps),
209 meshes, mesh-valued scalar fields (e.g. cortical thickness maps), deformation fields, scalar outputs
210 (e.g. volumes, regional averages), etc. Carrying out a neuroimaging study often involves the
211 combination of different pipelines or the chaining of a pipeline to another one. This is the case for
212 multimodal studies where processed outputs from a modality will be inputs for another pipeline, but it
213 is also true for studies involving a single modality: features extracted from one or several pipelines are
214 usually connected to statistical or machine learning frameworks. Finally, a structured organization for
215 processed data will ease the access and sharing of data, thus improving the reproducibility of
216 neuroimaging studies.

217 The CAPS format defines a hierarchy for the Clinica processed data. The idea is to include in
218 a single folder all the results generated by the different pipelines and to organize the data following the
219 main patterns of the BIDS specification. CAPS folders are kept separate from the raw data. Indeed,
220 when processing data, it is very common to have the raw dataset located on a separated storage or read-
221 only storage, while ongoing processed data are located on a separate location or on a faster data storage.

222 Another notion we often meet in neuroimaging studies is the notion of group, e.g. template
223 creation from a set of subjects or statistical analysis of a population. To handle these situations, we
224 simply add a level to the CAPS folder hierarchy. While pipeline outputs for individuals are stored in
225 the `subjects` folder, results of group studies are stored in the `groups` folder together with the set of
226 participants involved. For instance, an AD group label could be used when a template is created for a
227 group of Alzheimer's disease patients. Any time this AD template is used, the `group_label` is
228 provided to identify the pipeline outputs obtained for this group. The group `HCVsAD` could be used as
229 `group_label` for a statistical group comparison between healthy controls (HC) and Alzheimer's

230 disease patients. An illustration showing the chaining of pipelines and the creation of a group label can
231 be found in section 6.

232

233 **3.4 Clinica command line arguments**

234 For each pipeline, the command line interface will require a set of arguments which can be compulsory
235 or optional. The number of mandatory arguments is kept as small as possible, to ease its use. This set
236 of arguments is gathered into four categories.

237 First, the user will be asked to provide the Clinica mandatory arguments. These arguments are
238 in general the BIDS directory, the CAPS directory and/or the Group label, which were described in the
239 Clinica environment section (sections 3.2 and 3.3).

240 Then, several options are common to every pipeline: the Clinica standard options. For instance,
241 we can run a pipeline on a subset of participants and sessions by specifying a TSV file. Moreover, it is
242 possible to specify the number of cores of your machine used to run pipelines in parallel thanks to the
243 Nipype engine (Gorgolewski et al., 2011). A working directory can be specified for each pipeline. This
244 directory gathers all the inputs and outputs of the different steps of the pipeline, which is very useful
245 for debugging. It is especially useful in case a pipeline execution crashes to relaunch it with the exact
246 same parameters, allowing the execution to continue from the last successfully executed node.

247 Other parameters, specific to each pipeline, are gathered in the category “Optional parameters”.
248 For instance, when applying a smoothing filter with a specific full width at half maximum (FWHM),
249 this parameter can be specified.

250 Finally, advanced parameters for users with good knowledge of the pipeline itself or of the
251 software behind the pipeline will be gathered in the category “Advanced pipeline options”.

252

253 **3.5 List of atlases available in Clinica**

254 Depending on the modality studied and the type of analysis (voxel-based or surface-based), different
255 atlases can be used to generate regional features. These atlases are briefly listed below, and the reader
256 can refer to the documentation available on the website for further details.

257 When performing volumetric processing of T1w MRI and PET images, as done in the `t1-`
258 `volume*` and `pet-volume` pipelines, atlases defined in MNI space containing regions covering the
259 whole cortex and the main subcortical structures available are used (Samper-González et al., 2018),
260 currently AAL2 (Tzourio-Mazoyer et al., 2002), AICHA (Joliot et al., 2015), Hammers (Hammers et
261 al., 2003; Gousias et al., 2008), LPBA40 (Shattuck et al., 2008), and Neuromorphometrics¹⁴.

262 When running the `dwi-dti` pipeline, the JHU DTI81 (Wakana et al., 2007; Hua et al., 2008)
263 and JHUTracts[0|25|50] (Mori et al., 2005) atlases¹⁵, included in FSL (Jenkinson et al., 2012), defined

¹⁴ www.neuromorphometrics.com

¹⁵ <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases>

264 in MNI space, are used. JHU DTI81 contains 48 white matter tract labels and JHUTracts[0|25|50]
265 contains 20 white matter probabilistic tract labels with a 0%, 25% and 50% threshold.

266 Moreover, surface atlases are used when processing T1w MRI (respectively PET images) with
267 the `t1-freesurfer*` (respectively `pet-surface*`) pipelines. Currently, Clinica provides the
268 Desikan-Killiany (Desikan et al., 2006) atlas, which divides the cerebral cortex into gyri and contains
269 34 regions per hemisphere, and the Destrieux (Destrieux et al., 2010) atlas, which divides the cerebral
270 cortex into gyri and sulci and contains 74 regions per hemisphere.

271

272 3.6 Continuous Integration, testing and package distribution

273 The source code of the Clinica's platform follows the most standard current practices for software
274 development. The code is hosted in a publicly available platform¹⁶ and it uses a version control system.
275 A rigorous code review is performed for every contribution. The project has adopted a commonly used
276 workflow for development and the code is tested at different stages under controlled conditions. In
277 order to do this, several pipelines are executed by the continuous integration setup at different levels.

278 For each contribution proposal:

- 279 - The most recent commit pushed to the repository triggers a first iteration of the test suite. This
280 first round validates the package environment, the installation process and the correct
281 instantiation of the main tools proposed by Clinica.
- 282 - A draft of the documentation is written and published once the first iteration is over.

283 Then, the contribution proposal is reviewed and validated by a peer. Subsequently:

- 284 - Nightly tests ensure that new contributions do not introduce regressions in the results of the
285 software. This second iteration of the test suite runs the full set of Clinica's functionalities and,
286 due to the long processing time, they are executed once a day.
- 287 - Package construction and deployment is automatized by adding a tag with the version number
288 to the VCS. Versioned packages are published in the Python Package Index¹⁷.

289 The management of the continuous integration system is handled by a master server that creates the
290 link between the code repository and the different virtual machines that execute the continuous
291 integration tasks. Virtual machines are configured with Linux and macOS operating systems.

292 Outputs from the continuous integration process are publicly available and contributors can easily
293 consult them. Due to legal restrictions, the datasets used during the continuous integration cannot be
294 publicly distributed but detailed instructions on how to obtain them are provided on demand.

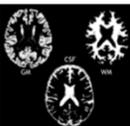
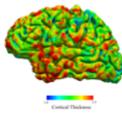
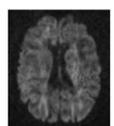
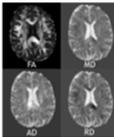
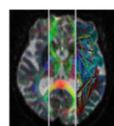
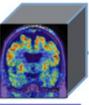
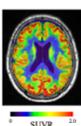
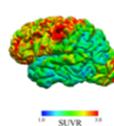
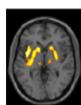
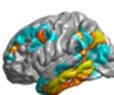
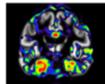
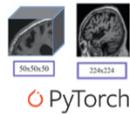
295

296

297

¹⁶ <https://github.com/aramis-lab/clinica/>

¹⁷ <https://pypi.org/project/clinica/>

Anatomical MRI	<p>t1-linear Bias field correction, affine registration and cropping Dependencies: ANTs</p> <ul style="list-style-type: none"> T1 MRI on ICBM 2009c nonlinear symmetric template Used as input for deeplearning-prepare-data  <p>169x208x179</p>	<p>t1-volume Tissue segmentation (GM, WM, CSF), inter-subject registration using Dartel, spatial normalization to standard space (MNI) Dependencies: SPM, CAT12</p> <ul style="list-style-type: none"> Voxel-based features (GM, WM, CSF) Regional features (average GM) using atlases (currently AALZ, AICHA, Hammers, LPBA40, Neuromorphometrics) 	<p>t1-freesurfer t1-freesurfer-Longitudinal Cortical surface extraction, segmentation of subcortical structures, cortical thickness estimation, spatial normalization to standard space (FsAverage) Dependencies: FreeSurfer</p> <ul style="list-style-type: none"> Surface-based features (cortical thickness) Regional features (average cortical thickness) using atlases (currently Desikan, Destrieux) 
Diffusion MRI	<p>dwi-preprocessing Correction of raw DWI data Dependencies: FSL, ANTs, Convert3D</p> <ul style="list-style-type: none"> EPI correction using phase-difference map fieldmap or T1w ("fieldmap-less") Prerequisite for dwi-dti or dwi-connectome pipelines 	<p>dwi-dti Extraction of DTI-based measures, normalization to standard space (MNI) Dependencies: FSL, ANTs, MRtrix3</p> <ul style="list-style-type: none"> Voxel-based features (FA, MD, AD, RD) Regional features (average FA, MD, AD, RD) using atlases (JHU DTI1, JHUTracts) 	<p>dwi-connectome Tractography & connectome Dependencies: FreeSurfer, FSL, MRtrix</p> <ul style="list-style-type: none"> Probabilistic tractography Structural connectome using atlases (currently Desikan, Destrieux) 
PET	<p>pet-linear Affine registration, intensity normalization and cropping Dependencies: ANTs</p> <ul style="list-style-type: none"> PET on ICBM 2009c nonlinear symmetric template Used as input for deeplearning-prepare-data  <p>169x208x179</p>	<p>pet-volume Registration to T1 MRI, partial volume correction, spatial normalization to standard space (MNI), intensity normalization Dependencies: SPM, PETPVC, CAT12</p> <ul style="list-style-type: none"> Voxel-based features (e.g. FDG uptake, amyloid uptake) Regional features (average FDG, amyloid uptake) using atlases (currently AALZ, AICHA, Hammers, LPBA40, Neuromorphometrics)  <p>SUVR</p>	<p>pet-surface pet-surface-Longitudinal Registration to T1 MRI, intensity normalization, partial volume correction, projection to cortical surface, spatial normalization to standard space (FsAverage) Dependencies: FreeSurfer, FSL, SPM, PETPVC</p> <ul style="list-style-type: none"> Surface-based features (e.g. FDG uptake, amyloid uptake) Regional features (average cortical thickness) using atlases (currently Desikan, Destrieux)  <p>SUVR</p>
Statistics	<p>statistics-volume Voxel-based mass-univariate analysis with SPM Dependencies: SPM, Matlab</p> <ul style="list-style-type: none"> Voxel-based features from t1-volume or pet-volume pipelines Group comparison using GLM 	<p>statistics-surface Surface-based mass-univariate analysis with SurfStat Dependencies: Matlab</p> <ul style="list-style-type: none"> Surface-based features from t1-freesurfer or pet-surface pipelines Group comparison or correlations analysis using GLM 	
Machine Learning	<p>machinelearning-prepare-spatial-svm Preparation of T1 MRI and PET data for spatially regularized SVM Dependencies: None</p> <ul style="list-style-type: none"> Regularization that accounts for the spatial and anatomical structure of neuroimaging data leading to a more regular and anatomically interpretable decision function. Used as input for machine learning classification 	<p>(No command line interface) Classification based on machine learning Dependencies: None</p> <ul style="list-style-type: none"> Voxel-based, surface-based or regional features Classifications (SVM, ℓ_2 logistic regression, random forests) using cross-validations (K-fold, repeated K-fold, repeated hold-out) 	
Deep Learning	<p>deeplearning-prepare-data Convert features extracted by Clinica to PyTorch tensors Dependencies: None</p> <ul style="list-style-type: none"> 3D images, 3D patches or 2D slices from t1-linear or t1-linear pipelines Tensors for PyTorch 	<p>clinicadi Reproducible classification of Alzheimer's disease using deep learning Installation: pip install clinicadi</p> <ul style="list-style-type: none"> Tensors from deeplearning-prepare-data pipeline CNN Classification using K-fold cross-validation or single data split Features such as random search, interpretability and meta-data management 	

298

299 **Figure 2** List of the pipelines currently available in Clinica with their dependencies and outputs.
 300 Explanations regarding the atlases can be found in section 3.5. GM: gray matter; CSF: cerebrospinal
 301 fluid; WM: white matter; FA: fractional anisotropy; MD: mean diffusivity; AD: axial diffusivity; RD:
 302 radial diffusivity, SVM: Support Vector Machine; ICBM: International Consortium for Brain
 303 Mapping.
 304

305 4 Image processing pipelines (clinica run)

306 This section gives a brief description of the different pipelines currently provided by Clinica as well as
307 the types of features Clinica can produce. An illustrative summary of the pipelines can be found in
308 **Erreur ! Source du renvoi introuvable.**

309 For technical details, we refer the reader to the online user documentation available on the
310 Clinica website where a longer description of each pipeline is provided.

311 4.1 Anatomical MRI

312 4.1.1 Linear processing of T1-weighted MR images (t1-linear)

313 The `t1-linear` pipeline performs a set of steps in order to affinely align T1w MR images to the MNI
314 space using the ANTs software package (Avants et al., 2014a, 201). These steps include: bias field
315 correction using N4ITK (Tustison et al., 2010); affine registration to the MNI152NLin2009cSym
316 template (Fonov et al., 2009, 2011) in MNI space with the SyN algorithm (Avants et al., 2008a);
317 cropping of the registered images to remove the background.

318 This pipeline was designed to be a prerequisite for the `deeplearning-prepare-data`
319 pipeline.

320

321 4.1.2 Processing of T1-weighted MR images for volume analyses using SPM (t1-volume*)

322 The `t1-volume*` pipelines extract voxel-based anatomical features from T1w MR images.
323 Specifically, they perform segmentation of tissues (gray matter (GM), white matter (WM),
324 cerebrospinal fluid (CSF)), normalization to MNI space and computation of regional measures using
325 atlases. Their main outputs are voxel-based maps of tissue density and average measures within cortical
326 regions stored as TSV files.

327 To that purpose, the pipeline wraps the Segmentation, Run Dartel and Normalise to MNI Space
328 routines implemented in SPM (Ashburner, 2012). First, the Unified Segmentation procedure
329 (Ashburner and Friston, 2005) is used to simultaneously perform tissue segmentation, bias field
330 correction and spatial normalization of the input image. Next, a group template is created using
331 DARTEL, an algorithm for diffeomorphic image registration (Ashburner, 2007), from the subjects'
332 tissue probability maps on the native space, usually GM, WM and CSF, obtained at the previous step.
333 The DARTEL to MNI method (Ashburner, 2007) is then applied, providing a registration of the native
334 space images into the MNI space.

335

336 4.1.3 Processing of T1-weighted MR images for surface analyses using FreeSurfer (t1-freesurfer; 337 t1-freesurfer-longitudinal)

338 The `t1-freesurfer` pipeline is mainly a wrapper of the `recon-all` tool of FreeSurfer (Fischl,
339 2012). It performs segmentation of subcortical structures (Fischl et al., 2002, 2004a), extraction of
340 cortical surfaces, cortical thickness estimation (Fischl and Dale, 2000), spatial normalization onto the
341 FreeSurfer surface template (FsAverage) (Fischl et al., 1999), and parcellation of cortical regions using
342 the Desikan and Destrieux atlases (Fischl et al., 2004b). Its main outputs are surface-based cortical
343 thickness features and regional statistics (e.g. regional volume, mean cortical thickness).

344 The `t1-freesurfer-longitudinal` pipeline processes a series of images acquired at
345 different time points for the same subject with the longitudinal FreeSurfer stream (Reuter et al., 2012)
346 to increase the accuracy of volume and thickness estimates. It does so in a single command consisting
347 of two consecutive steps: 1) within-subject template creation (`recon-all -base` command) to
348 produce an unbiased template image from the different time points using robust and inverse consistent
349 registration (Reuter et al., 2010); 2) longitudinal correction (`recon-all -long` command):
350 segmentation, surface extraction and computation of measurements at each time point.

351

352 4.2 Diffusion MRI

353 4.2.1 DWI pre-processing (`dwi-preprocessing-*`)

354 The `dwi-preprocessing-*` pipelines correct diffusion-weighted MRI (DWI) datasets for motion,
355 eddy current, magnetic susceptibility and bias field distortions, assuming that the data have been
356 acquired using an echo-planar imaging (EPI) sequence.

357 Due to the heterogeneity in acquisitions of fieldmaps and techniques to correct magnetic
358 susceptibility distortions, several pipelines are proposed. Currently, Clinica can handle DWI datasets
359 with fieldmap data containing a phase-difference map (case ‘phase-difference map and at least one
360 magnitude image’ in the BIDS specifications¹⁸) (`dwi-preprocessing-using-fmap`) and DWI
361 datasets with no extra data (`dwi-preprocessing-using-t1`), which is the case of the public
362 Alzheimer’s Disease Neuroimaging Initiative (ADNI)¹⁹ dataset for instance.

363 In all cases, motion and eddy motion corrections are performed with the FSL software
364 (Jenkinson et al., 2012) using the `eddy` tool (Andersson and Sotiropoulos, 2016) with the `replace`
365 `outliers (--repol)` option (Andersson et al., 2016) while bias field is corrected with the ANTs N4 bias
366 correction (Tustison et al., 2010). Regarding susceptibility correction, the FSL `prelude/fugue` tools
367 were used for the `dwi-preprocessing-using-fmap` pipeline and the ANTs SyN registration
368 algorithm (Leow et al., 2007; Avants et al., 2008a) for the `dwi-preprocessing-using-t1` pipeline.
369 The outputs of the pipelines are the corrected DWI datasets and a brain mask of the `b=0` image.

370 These pipelines are prerequisites for the `dwi-dti` and `dwi-connectome` pipelines.

371

372 4.2.2 Computation of DTI, DTI-scalar maps and ROI analysis (`dwi-dti`)

373 The `dwi-dti` pipeline extracts voxel-based features from diffusion tensor imaging (DTI), namely the
374 fractional anisotropy (FA), mean diffusivity (MD), axial diffusivity (AD) and radial diffusivity (RD)
375 using MRtrix3 (Tournier et al., 2012). Then, the DTI-derived scalar maps (FA, MD, AD, RD) are
376 normalized with ANTs (Avants et al., 2008a) onto an FA-atlas with labelled tracts. Its main outputs
377 are voxel-based maps from DTI and average measures within tracts stored as TSV files.

378

¹⁸ <http://bids.neuroimaging.io/>

¹⁹ <http://adni.loni.usc.edu/>

379 **4.2.3 Computation of fiber orientation distributions, tractogram and structural connectome** 380 **(dwi-connectome)**

381 The `dwi-connectome` pipeline computes a weighted graph encoding anatomical connections between
382 a set of brain regions from corrected DWI datasets. To that aim, it relies on the MRtrix3 (Tournier et
383 al., 2012) software to compute the constrained spherical deconvolution diffusion model (Tournier et
384 al., 2007), perform probabilistic tractography (Tournier et al., 2010) and computes a connectome using
385 the Desikan & Destrieux atlases from FreeSurfer (Fischl, 2012). Its main outputs are the diffusion
386 model, the whole-brain tractography and the connectivity matrices.

387

388 **4.3 Positron emission tomography**

389 Currently, Clinica is supporting amyloid and FDG PET data but other tracers will be added in the
390 future.

391 **4.3.1 Linear processing of PET images (pet-linear)**

392 The `pet-linear` pipeline performs a spatial normalization to the MNI space and intensity
393 normalization of PET images. The first step of the pipeline is an affine registration to the
394 MNI152NLin2009cSym template (Fonov et al., 2009, 2011) in MNI space with the SyN algorithm
395 (Avants et al., 2008b) from the ANTs software package (Avants et al., 2014b). Then, the registered
396 image intensity is normalized using the mean intensity in reference regions resulting in a standardized
397 uptake value ratio (SUVR) map. The normalized imaged is finally cropped to remove the background.

398

399 **4.3.2 Processing of PET images for volume analyses (pet-volume)**

400 The `pet-volume` pipeline extracts voxel-based features from PET data. Specifically, it performs intra-
401 subject registration of the PET image into the space of the subject's T1w MR image using SPM
402 (Ashburner, 2012). Optionally, partial volume correction (PVC) can be applied thanks to the PETPVC
403 toolbox (Thomas et al., 2016). Then, inter-subject spatial normalization of the PET image into MNI
404 space is performed based on the DARTEL deformation model of SPM (Ashburner, 2007) and intensity
405 normalization is done using the average PET uptake in a reference region resulting in a SUVR map.
406 Its main outputs are voxel-based maps of SUVR and average measures within cortical regions.

407

408 **4.3.3 Processing of PET images for surface analyses (pet-surface; pet-surface-longitudinal)**

409 The `pet-surface` pipeline extracts the PET signal and projects it onto the cortical surface using the
410 approach described in (Marcoux et al., 2018). More precisely, it performs co-registration of PET and
411 T1w MRI, intensity normalization, PVC with the PETPVC toolbox (Thomas et al., 2016), robust
412 projection of the PET signal onto the subject's cortical surface, parcellation of the cortical regions
413 using the Desikan and Destrieux atlases and spatial normalization onto the FreeSurfer (Fischl, 2012)
414 surface template (FsAverage). Its main outputs are surface-based PET uptake and regional statistics
415 (mean PET uptake) stored as TSV files.

416 The `pet-surface-longitudinal` pipeline performs the same steps as the `pet-surface`
417 pipeline except that the cortical and white surfaces are estimated with the longitudinal pipeline of
418 FreeSurfer (Reuter et al., 2012).

419

420 **4.4 Statistics**

421 **4.4.1 Voxel-based mass-univariate analysis with SPM (statistics-volume)**

422 The `statistics-volume` pipeline performs statistical analysis on volume-based features using the
423 general linear model (GLM) and random field theory (Worsley et al., 2009). To that aim, the pipeline
424 wraps the statistical analysis toolbox implemented in SPM. Volume-based measurements can be gray
425 matter maps from the `t1-volume` pipeline or PET measurements from the `pet-volume` pipeline.
426 Currently, statistical analysis includes only group comparison.

427

428 **4.4.2 Surface-based mass-univariate analysis with SurfStat (statistics-surface)**

429 The `statistics-surface` pipeline performs statistical analysis on surface-based features using the
430 GLM. To that aim, the pipeline relies on the Matlab toolbox SurfStat designed for statistical analyses
431 of univariate and multivariate surface and volumetric data using the GLM (Worsley et al., 2009).
432 Surface-based measurements are analyzed on the `FsAverage` surface template from FreeSurfer. The
433 pipeline can handle cortical thickness from the `t1-freesurfer` pipeline or PET measurements from
434 the `pet-surface` pipeline. Currently, statistical analysis includes group comparison and correlation.

435

436 **4.5 Machine Learning**

437 **4.5.1 Classification based on machine learning (no command line)**

438 Clinica provides a modular way to perform classification based on machine learning. To build their
439 own classification pipeline, the user can combine three modules based on the scikit-learn library
440 (Pedregosa et al., 2011):

- 441 - Input module. The user can select the inputs from the features available in the CAPS directory,
442 such as gray matter maps obtained from T1w MR images, or SUVR maps obtained from FDG
443 PET images.
- 444 - Algorithm module. The user can choose between different classifiers, currently support vector
445 machine (SVM), logistic regression and random forest.
- 446 - Validation module. Several cross-validation (CV) methods are available: k-fold CV, repeated
447 k-fold CV and repeated hold-out CV.

448 Note that no command line interface is available for these specific tools.

449 More details regarding the different modules and a description of the way they can be used to
450 perform reproducible evaluation of classification methods in Alzheimer’s disease can be found in
451 (Samper-González et al., 2018) and its dedicated repository²⁰.

452

453 **4.5.2 Spatially-regularized support vector machine (machinelearning-prepare-spatial-svm)**

454 The machinelearning-prepare-spatial-svm pipeline allows the preparation of T1w MRI and
455 PET data to perform classification with an SVM with spatial and anatomical regularization (Cuingnet
456 et al., 2013). In this approach, the standard regularization of the SVM is replaced with a regularization
457 that accounts for the spatial and anatomical structure of neuroimaging data. More specifically, it is
458 regularized with respect to the tissue maps (GM, WM, CSF). As a result, the decision function learned
459 by the algorithm will be more regular and anatomically interpretable. Because the SVM is a kernel
460 method, the spatial/anatomical regularization is done as a pre-processing on the feature maps and the
461 result can then be fed to a standard linear SVM.

462

463 **4.6 Deep Learning**

464 **4.6.1 Prepare data for deep learning (deeplearning-prepare-data)**

465 The deeplearning-prepare-data pipeline allows the preparation of data for subsequent training
466 or inference of deep learning models. To that aim, it uses the outputs from `t1-linear` or `pet-linear`
467 pipelines. Specifically, 3D images, 3D patches or 2D slices can be extracted and converted into
468 PyTorch tensors (Paszke et al., 2019).

469

470 **4.6.2 Training and validation of deep learning models (clinicadl)**

471 The training and validation of deep learning models based on Clinica outputs can be performed using
472 a dedicated Python library: ClinicaDL²¹. This extension of Clinica contains essential features for deep
473 learning application to 3D medical images:

- 474 - modules to split data avoiding data leakage, which is a major problem in the domain (Wen et
475 al., 2020);
- 476 - a training method for autoencoders, CNN and a multi-CNN framework which allows the use
477 of other networks trained with ClinicaDL for transfer learning;
- 478 - a testing function to evaluate the performance of classifiers on independent test sets;
- 479 - saliency maps generation (Simonyan et al., 2013) extensively used to interpret the outputs of
480 deep learning networks;
- 481 - basic network architecture search tools, such as random search utilities and methods to generate
482 trivial synthetic datasets for architecture debugging.

²⁰ <https://github.com/aramis-lab/AD-ML>

²¹ <https://github.com/aramis-lab/AD-DL>

483 This library is documented in an independent documentation²². An online tutorial²³ allows beginners
484 to better understand these functionalities by testing them locally, or on Google Colab if they do not
485 have access to sufficient computational resources.

486

487 **5 Clinica utilities**

488 **5.1 Conversion of neuroimaging datasets into a BIDS hierarchy (clinica convert)**

489 Clinica provides tools to curate several publicly available neuroimaging datasets and automatically
490 convert them into the BIDS standardized data structure. This section explains what the user needs to
491 download prior to running the converter and the rationale behind the selection of data when multiple
492 acquisitions or pre-processing steps are available. For all converters, the user only needs to download
493 the dataset. All subsequent conversion steps are performed automatically (no user intervention is
494 required) and use parallelization for faster processing. For further details, the reader can refer to
495 (Samper-González et al., 2018).

496

497 **5.1.1 Conversion of the ADNI dataset to BIDS (adni-to-bids)**

498 The ADNI to BIDS converter requires the user to have downloaded all the ADNI study data (tabular
499 data in CSV format) and the imaging data of interest. Note that the downloaded files must be kept
500 exactly as they were downloaded. The imaging modalities currently being converted to BIDS include
501 T1w MRI, FLAIR, DWI, fMRI, FDG PET, PiB PET, Florbetapir (AV45) PET and Flortaucipir
502 (AV1451) PET. Clinical data are also converted to BIDS. They include data that do not change over
503 time, such as the subject's sex, education level or diagnosis at baseline, as well as session-dependent
504 data, such as the clinical scores. The clinical data being converted are defined in a spreadsheet that is
505 available with the code of the converter. The user can easily modify this file if they want to convert
506 additional clinical data.

507

508 **5.1.2 Conversion of the AIBL dataset to BIDS (aibl-to-bids)**

509 The AIBL to BIDS converter requires the user to have downloaded the AIBL non-imaging data (tabular
510 data in CSV format) and the imaging data of interest. For each AIBL participant, the T1w MRI and the
511 Florbetapir, PiB and Flutemetamol PET images are converted. As for the ADNI converter, clinical data
512 converted to BIDS are defined in a spreadsheet available with the code of the converter, which the user
513 can modify.

514

515 **5.1.3 Conversion of the NIFD dataset to BIDS (nifd-to-bids)**

516 The NIFD to BIDS converter requires the user to have downloaded the NIFD imaging data alongside
517 the corresponding clinical data in CSV format. For each NIFD participant, the T1w MRI, FLAIR, PiB

²² <https://clinicadl.readthedocs.io/en/latest/>

²³ <https://aramislab.paris.inria.fr/clinicadl/tuto/intro.html>

518 PET and FDG PET images are converted. The clinical data conversion is as described in the previous
519 sections.

520

521 **5.1.4 Conversion of the OASIS dataset to BIDS (oasis-to-bids)**

522 The OASIS to BIDS converter requires the user to have downloaded the OASIS-1 imaging data and
523 the associated CSV file. For each subject, among the multiple T1w MR images available, we select the
524 average of the motion-corrected co-registered individual images resampled to 1 mm isotropic voxels.
525 The clinical data are converted as described in the previous sections.

526

527 **5.1.5 Syntax to run the converters**

528 After having downloaded the clinical and imaging data of one of these studies, the conversion of a
529 dataset into BIDS is performed using the following syntax:

```
530 clinica convert <dataset>-to-bids dataset_directory clinical_data_directory  
531 bids_directory
```

532 where <dataset>-to-bids can be adni-to-bids, aibl-to-bids, nifd-to-bids or oasis-
533 to-bids.

534

535 **5.2 Data handling tools (clinica iotools)**

536 We also propose a set of tools that allows the user to handle BIDS and CAPS datasets. For the moment,
537 there are five different commands:

- 538 - `center-nifti`: Center Nifti files of a BIDS directory.
- 539 - `check-missing-modalities`: This command checks missing modalities in a BIDS
540 directory.
- 541 - `check-missing-processing`: This command checks the outputs in a CAPS directory.
- 542 - `create-subjects-visits`: This command generates a list of subjects with their sessions
543 based on a BIDS directory and stores the outputs in a TSV file.
- 544 - `merge-tsv`: This command merges all the tabular data including the clinical data of a BIDS
545 directory and the regional features from a CAPS directory (e.g. mean GM density in AAL2
546 atlas) into a single TSV file. This file can then be easily plugged into machine learning tools
547 via Clinica or other statistical/machine learning software packages.

548

549 **6 Usage example**

550 In this section, we propose to show how Clinica can be used to perform a group comparison of FDG
 551 PET data projected on the cortical surface between patients with Alzheimer’s disease and healthy
 552 controls from the ADNI database. An illustrative summary of this example can be found in Figure 3.

553 To download the ADNI dataset, it is necessary to register to the LONI Image & Data Archive²⁴,
 554 a secure research data repository, and request access to the ADNI dataset through the submission of an
 555 online application form. Both the imaging and clinical data need to be downloaded, each to a folder
 556 that we will call `imaging_data_dir` and `clinical_data_dir`, respectively. The following
 557 command can be used to convert the T1 and FDG PET data of the ADNI dataset into BIDS:

```
558 clinica convert adni-to-bids imaging_data_dir clinical_data_dir ADNI_BIDS
559 --modalities T1 PET_FDG
```

560 where the `ADNI_BIDS` folder contains the conversion of ADNI into BIDS. We can now start processing
 561 the data. First, we need to extract the cortical surfaces from each anatomical image. To do so, we simply
 562 need to type on the terminal the following command:

```
563 clinica run t1-freesurfer ADNI_BIDS ADNI_CAPS
```

564 where the output data will be stored in the `ADNI_CAPS` folder. After visual inspection of the generated
 565 outputs, the FDG PET data can be projected onto the cortex. The command line will be:

```
566 clinica run pet-surface ADNI_BIDS ADNI_CAPS fdg pons pvc_psf.tsv
```

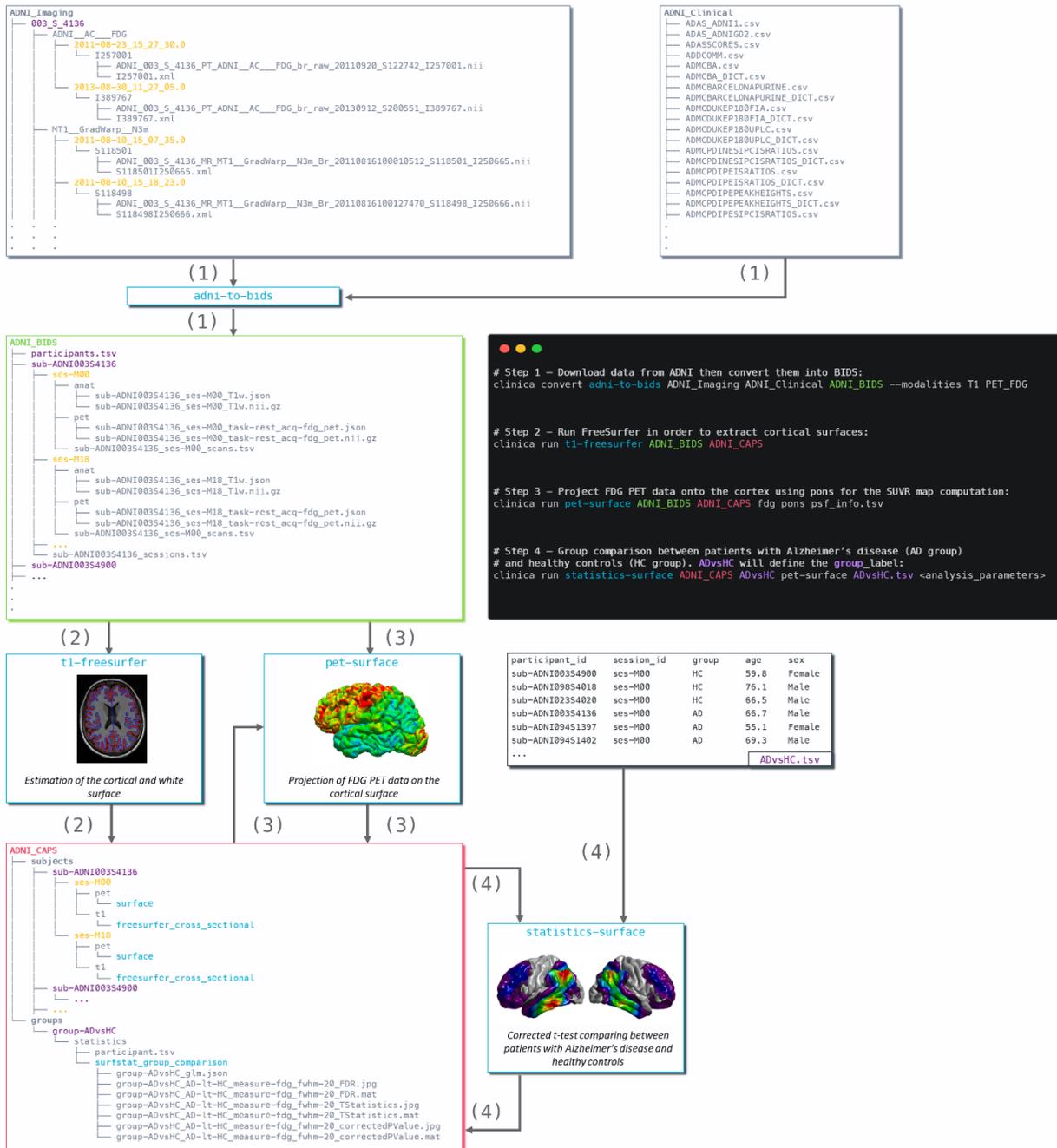
567 where `fdg` is the label given to the PET acquisition, `pons` is the reference region for the SUVR map
 568 computation and `pvc_psf.tsv` is the TSV file containing PSF information for each PET image.
 569 Finally, we can perform group comparison of cortical FDG PET data after having checked the outputs.
 570 The demographic information of the population studied will be stored in a TSV file, looking as follows:

```
571 participant_id    session_id    group    age    sex
572 sub-ADNI094S2201 ses-M00    HC    63.7    Female
573 sub-ADNI098S4018 ses-M00    HC    76.1    Male
574 sub-ADNI023S4020 ses-M00    HC    66.5    Male
575 sub-ADNI031S4021 ses-M00    HC    66.5    Male
576 sub-ADNI094S1397 ses-M00    AD    55.1    Female
577 sub-ADNI094S1402 ses-M00    AD    69.3    Male
578 sub-ADNI128S1409 ses-M00    AD    65.9    Male
579 sub-ADNI128S1430 ses-M00    AD    83.4    Female
580 ...
```

581 where participants with Alzheimer’s disease (respectively healthy controls) have the AD label
 582 (respectively HC label) in the `group` column. We will call this file `ADvsHC_participants.tsv`.
 583 Using age and sex as covariates, the command line will be:

²⁴ <https://ida.loni.usc.edu>

```
584 clinica run statistics-surface ADNI_CAPS ADvsHC pet-surface \  
585     group_comparison ADvsHC_participants.tsv group -covariates "age + sex"  
586     The results of the statistical analysis will be stored in the ADNI_CAPS/groups/group-  
587     ADvsHC folder.
```



588

589 **Figure 3** Diagram illustrating the Clinica pipelines involved when performing a group comparison of
 590 FDG PET data projected on the cortical surface between patients with Alzheimer’s disease and healthy
 591 controls from the ADNI database. First, clinical and neuroimaging data are downloaded from the ADNI
 592 website and data are converted into BIDS with the `adni-to-bids` tool from Clinica (1). Estimation
 593 of the cortical and white surface is then produced by the `t1-freesurfer` pipeline in a single
 594 command line (2). Afterwards, FDG PET data can be projected on the subject’s cortical surface and
 595 normalized to the FsAverage template from FreeSurfer using the `pet-surface` pipeline (3). Finally,
 596 a TSV file with demographic information of the population studied is given to the `statistics-`
 597 `surface` pipeline to generate the results of the group comparison between patients with Alzheimer’s
 598 disease and healthy controls (4).

599 7 Discussion

600 We proposed a software platform that aims at making clinical neuroscience easier and more
601 reproducible. Clinica automates the processing of pipelines involving several neuroimaging modalities
602 (currently, anatomical MRI, diffusion MRI, and PET) as well as statistics, machine learning and deep
603 learning tools. Additionally, Clinica provides tools to convert public neuroimaging datasets focused on
604 dementia (ADNI, AIBL, OASIS and NIFD) into the BIDS standard, and tools to handle raw (BIDS)
605 and processed datasets. The use of the BIDS standard as the only prerequisite on the data and the
606 unified command line interface across the pipelines ease the processing automation. The image analysis
607 automation is also improved by the use of the CAPS hierarchy, which facilitates the chaining of
608 pipelines. The target audience of Clinica is neuroscientists or clinicians conducting clinical
609 neuroscience studies involving multimodal imaging, and researchers developing advanced machine or
610 deep learning algorithms.

611 The last three decades witnessed the development of many software packages for the processing
612 of neuroimaging data. A first category of packages comprises those implementing innovative image
613 processing methodologies (e.g. tissue segmentation, registration). Many tools fall into this category,
614 for instance SPM (Friston et al., 2007), AFNI (Cox, 1996), FreeSurfer (Fischl, 2012), FSL (Jenkinson
615 et al., 2012), PETPVC (Thomas et al., 2016), Camino (A Cook et al., 2005), Dipy (Garyfallidis et al.,
616 2014), DTI-TK²⁵, MRtrix (Tournier et al., 2012) or ANTs (Avants et al., 2014a). Some of these tools
617 cover a variety of modalities while others focus on a specific one (diffusion MRI for Camino, Dipy,
618 DTI-TK and MRtrix, fMRI for AFNI, PET for PETPVC). However, performing a multimodal study
619 can be difficult because one needs to combine tools from different packages. This results in complex
620 pipelines which can be difficult to build, maintain and distribute. Even when analyzing a single
621 modality, one often wants to combine tools from different packages, thus facing similar difficulties.
622 Combination of tools is made even more difficult by the fact that the input and output data are organized
623 differently by each tool.

624 Efforts of the community have alleviated several of these difficulties. The NeuroDebian
625 community²⁶ (Halchenko and Hanke, 2012) aims to provide and ease the installation of a large
626 collection of software packages for the Debian distribution. The Nipype (Gorgolewski et al., 2011)
627 system facilitates the building of complex pipelines through the wrapping of tools in Python. BIDS
628 (Gorgolewski et al., 2017) provides a standard for organizing data. BIDS-Apps provides versions of
629 software packages using BIDS for data organization. More generally, the NIPY community aims to
630 provide a comprehensive set of tools for the analysis of neuroimaging data in a single language, Python.
631 However, many useful tools and packages remain outside of the NIPY scope, being written in different
632 languages. As mentioned above, Nipype allows wrapping these heterogeneous tools. It is a powerful
633 and particularly useful tool for that aim. However, the building of pipelines remains left to the user.
634 This requires substantial development efforts. Clinicians/neuroscientists often do not have the
635 necessary programming expertise, while researchers in machine learning often do not have the
636 necessary neuroimaging expertise. Therefore, Nipype and Clinica do not have the same objectives.
637 Nipype provides a powerful way to write pipelines. As such, it is particularly flexible but requires some
638 programming expertise. Clinica, on the other hand, offers a set of predefined pipelines: it is thus easier
639 to use but less flexible.

²⁵ <http://dti-tk.sourceforge.net>

²⁶ <http://neuro.debian.net/>

640 There are also software packages that integrate different tools within a single environment. This
641 is for example the case of BCBtoolkit (Foulon et al., 2018), BrainVISA (Cointepas et al., 2001),
642 BrainSuite²⁷, fMRIPrep (Esteban et al., 2019), or Pypes (Savio et al., 2017). Clinica falls within this
643 category. It shares some characteristics with these tools but also has important differences. The
644 BCBtoolkit wraps neuroimaging software packages from the community but also highlights new
645 methodological developments to evaluate brain disconnections. BCBookit does not use any pipelining
646 system but instead wraps bash scripts that are then made available through a GUI. The BrainVISA
647 platform, even though it also wraps some existing tools, mainly provides innovative tools for the
648 analysis of human or animal brain imaging data. Moreover, it includes its own pipelining system while
649 Clinica relies on the community effort Nipype. BrainSuite does not wrap existing tools but provides a
650 set of innovative tools for the analysis of neuroimaging data. It can be executed using a GUI, a
651 command line, a Nipype interface, or as a BIDS App. fMRIPrep combines software components using
652 Nipype (Gorgolewski et al., 2011) to provide a robust pipeline for pre-processing of fMRI data. It
653 assumes input data to follow the BIDS standard (Gorgolewski et al., 2016) and outputs are organized
654 following the ongoing BIDS-derivatives initiative. Pypes is probably the closest in spirit to Clinica: it
655 also focuses on the integration of existing tools into a set of reusable pipelines built with Nipype. The
656 user needs to specify a configuration file to describe the input data even for BIDS datasets. The output
657 data will then follow the same structure as the input data and can be chained to other pipelines from
658 Pypes. Note that the ability to chain pipelines exists for a limited number of pipelines in the BIDS-App
659 version of BrainSuite and is likely to be more present in BIDS Apps with the advent of BIDS-
660 derivatives. Although Clinica does not provide a GUI, efforts were made to simplify as much as
661 possible its command line interface, which is feasible thanks to the autocompletion, the structured
662 organization of data and the documentation which was designed in order to be readable by a newcomer.

663 Machine learning is now widely used in the neuroimaging community, for cognitive
664 neuroscience or computer-aided diagnosis applications. However, applying such approaches to
665 neuroimaging data can be difficult for newcomers. Conversely, researchers in machine learning are
666 often interested in applying and validating their approaches to clinical neuroimaging problems (e.g.
667 diagnosis, prognosis). However, they often lack the necessary knowledge for preprocessing
668 neuroimaging data and extracting features. Nilearn (Abraham et al., 2014) has allowed major progress
669 in that direction by providing in a single environment tools for preprocessing, data manipulation,
670 feature extraction and machine learning wrapping the scikit-learn library (Pedregosa et al., 2011).
671 However, Nilearn currently mostly targets cognitive neuroimaging and mainly deals with functional
672 neuroimaging. On the other hand, Clinica is dedicated to clinical neuroimaging studies, such as
673 biomarker design and clinical decision support systems. As a result, Clinica does not aim to deal with
674 task-based fMRI and is currently mostly focused on the analysis of T1w MRI, diffusion MRI and PET
675 data. Deep learning methods are also increasingly applied to neuroimaging. Even though such methods
676 usually do not rely on pre-extracted features, they will still require some preprocessing and data
677 conversion tools in order to use neuroimaging data in frameworks such as PyTorch. Clinica provides
678 such tools.

679 Reproducibility has been highlighted as a major challenge in many scientific fields including
680 neuroimaging (Poldrack et al., 2017). This problem has also been highlighted in machine learning for
681 healthcare in general (McDermott et al., 2019) and for brain diseases in particular (Samper-González
682 et al., 2018; Wen et al., 2021). Clinica aims to make reproducible research easier to perform. To that
683 purpose, it combines: 1) use of a community standard for inputs; 2) the definition of a standardized
684 organization for outputs; 3) standardized ways to extract features; 4) extensive software testing. As

²⁷ <http://brainsuite.org>

685 previously mentioned, it extensively relies on community achievements such as the BIDS standard and
686 Nipype.

687 Clinica currently has the following limitations. First, we will improve the enforcement of
688 reproducibility by adding traceability features. Furthermore, quality control (QC) of processed data is
689 currently done using standard image viewers which is clearly suboptimal. We plan to add more
690 advanced QC of outputs in the spirit for instance of MindControl (Keshavan et al., 2018). Finally, new
691 longitudinal analysis pipelines (beyond those already present for FreeSurfer and PET surface
692 processing) will be developed.

693

694 8 Conflict of Interest

695 The authors declare that the research was conducted in the absence of any commercial or financial
696 relationships that could be construed as a potential conflict of interest.

697 9 Author Contributions

698 Study concepts/study design, all authors; manuscript drafting or manuscript revision for important
699 intellectual content, all authors; approval of final version of submitted manuscript, all authors; literature
700 research, AR, NB, OC; and manuscript editing, all authors.

701 10 Funding

702 The research leading to these results has received funding from the programs “Investissements
703 d’avenir” ANR-10-IAIHU-06 (Agence Nationale de la Recherche-10-IA Institut Hospitalo-
704 Universitaire-6), ANR-11-IDEX-004 (Agence Nationale de la Recherche-11- Initiative d’Excellence-
705 004, project LearnPETMR number SU-16-R-EMR-16) and ANR-19-P3IA-0001 (PRAIRIE 3IA
706 Institute), from the European Union H2020 program (project EuroPOND, grant number 666992,
707 project HBP SGA1 grant number 720270), from the joint NSF/NIH/ANR program “Collaborative
708 Research in Computational Neuroscience” (project HIPLAY7, grant number ANR-16-NEUC-0001-
709 01), from Agence Nationale de la Recherche (project PREVDEMALS, grant number ANR-14-CE15-
710 0016-07), from the ICM Big Brain Theory Program (project DYNAMO), from the Inria Project Lab
711 Program (project Neuromarkers), from the European Research Council (to Dr Durrleman project
712 LEASP, grant number 678304), and from the “Contrat d’Interface Local” program (to Dr Colliot) from
713 Assistance Publique-Hôpitaux de Paris (AP-HP). N.B. received funding from the People Programme
714 (Marie Curie Actions) of the European Union’s Seventh Framework Programme (FP7/2007-2013)
715 under REA grant agreement no. PCOFUND-GA-2013-609102, through the PRESTIGE programme
716 coordinated by Campus France.

717

718 **11 References**

- 719 A Cook, P., Bai, Y., Nedjati-Gilani, S., K Seunarine, K., Hall, M., Parker, G., et al. (2005). Camino:
720 Open-Source Diffusion-MRI Reconstruction and Processing. *Proc. Intl. Soc. Magn. Reson. Med.* 14.
- 721 Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., et al. (2014).
722 Machine learning for neuroimaging with scikit-learn. *Front. Neuroinform.* 8.
723 doi:10.3389/fninf.2014.00014.
- 724 Andersson, J. L. R., Graham, M. S., Zsoldos, E., and Sotiropoulos, S. N. (2016). Incorporating outlier
725 detection and replacement into a non-parametric framework for movement and distortion correction of
726 diffusion MR images. *NeuroImage* 141, 556–572. doi:10.1016/j.neuroimage.2016.06.058.
- 727 Andersson, J. L. R., and Sotiropoulos, S. N. (2016). An integrated approach to correction for off-
728 resonance effects and subject movement in diffusion MR imaging. *Neuroimage* 125, 1063–1078.
729 doi:10.1016/j.neuroimage.2015.10.019.
- 730 Ashburner, J. (2007). A fast diffeomorphic image registration algorithm. *NeuroImage* 38, 95–113.
731 doi:10.1016/j.neuroimage.2007.07.007.
- 732 Ashburner, J. (2012). SPM: A history. *Neuroimage* 62–248, 791–800.
733 doi:10.1016/j.neuroimage.2011.10.025.
- 734 Ashburner, J., and Friston, K. J. (2005). Unified segmentation. *NeuroImage* 26, 839–851.
735 doi:10.1016/j.neuroimage.2005.02.018.
- 736 Avants, B. B., Epstein, C. L., Grossman, M., and Gee, J. C. (2008a). Symmetric diffeomorphic image
737 registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative
738 brain. *Medical Image Analysis* 12, 26–41. doi:10.1016/j.media.2007.06.004.
- 739 Avants, B. B., Epstein, C. L., Grossman, M., and Gee, J. C. (2008b). Symmetric diffeomorphic image
740 registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative
741 brain. *Medical Image Analysis* 12, 26–41. doi:10.1016/j.media.2007.06.004.
- 742 Avants, B. B., Tustison, N. J., Stauffer, M., Song, G., Wu, B., and Gee, J. C. (2014a). The Insight
743 ToolKit image registration framework. *Front Neuroinform* 8. doi:10.3389/fninf.2014.00044.
- 744 Avants, B. B., Tustison, N. J., Stauffer, M., Song, G., Wu, B., and Gee, J. C. (2014b). The Insight
745 ToolKit image registration framework. *Front. Neuroinform.* 8. doi:10.3389/fninf.2014.00044.
- 746 Brett, M., Hanke, M., Markiewicz, C., Marc-Alexandre Côté, McCarthy, P., Cheng, C., et al. (2019).
747 *nipy/nibabel: 2.3.3*. Zenodo doi:10.5281/zenodo.2541736.
- 748 Cointepas, Y., Mangin, J.-F., Garnero, L., Poline, J.-B., and Benali, H. (2001). BrainVISA: Software
749 platform for visualization and analysis of multi-modality brain data. *NeuroImage* 13, 98.
750 doi:10.1016/S1053-8119(01)91441-7.
- 751 Cox, R. W. (1996). AFNI: software for analysis and visualization of functional magnetic resonance
752 neuroimages. *Comput. Biomed. Res.* 29, 162–173.
- 753 Cuingnet, R., Glaunès, J. A., Chupin, M., Benali, H., Colliot, O., and Alzheimer’s Disease
754 Neuroimaging Initiative (2013). Spatial and Anatomical Regularization of SVM: A General
755 Framework for Neuroimaging Data. *IEEE Trans Pattern Anal Mach Intell* 35, 682–696.
756 doi:10.1109/TPAMI.2012.142.
- 757 Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., et al. (2006). An
758 automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based

- 759 regions of interest. *NeuroImage* 31, 968–980. doi:10.1016/j.neuroimage.2006.01.021.
- 760 Destrieux, C., Fischl, B., Dale, A., and Halgren, E. (2010). Automatic parcellation of human cortical
761 gyri and sulci using standard anatomical nomenclature. *Neuroimage* 53, 1–15.
762 doi:10.1016/j.neuroimage.2010.06.010.
- 763 Esteban, O., Markiewicz, C. J., Blair, R. W., Moodie, C. A., Isik, A. I., Erramuzpe, A., et al. (2019).
764 fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nat Methods* 16, 111–116.
765 doi:10.1038/s41592-018-0235-4.
- 766 Fischl, B. (2012). FreeSurfer. *NeuroImage* 62, 774–781. doi:10.1016/j.neuroimage.2012.01.021.
- 767 Fischl, B., and Dale, A. M. (2000). Measuring the thickness of the human cerebral cortex from
768 magnetic resonance images. *PNAS* 97, 11050–11055. doi:10.1073/pnas.200033797.
- 769 Fischl, B., Salat, D. H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., et al. (2002). Whole Brain
770 Segmentation: Automated Labeling of Neuroanatomical Structures in the Human Brain. *Neuron* 33,
771 341–355. doi:10.1016/S0896-6273(02)00569-X.
- 772 Fischl, B., Salat, D. H., van der Kouwe, A. J. W., Makris, N., Ségonne, F., Quinn, B. T., et al. (2004a).
773 Sequence-independent segmentation of magnetic resonance images. *NeuroImage* 23, S69–S84.
774 doi:10.1016/j.neuroimage.2004.07.016.
- 775 Fischl, B., Sereno, M. I., and Dale, A. M. (1999). Cortical Surface-Based Analysis: II: Inflation,
776 Flattening, and a Surface-Based Coordinate System. *NeuroImage* 9, 195–207.
777 doi:10.1006/nimg.1998.0396.
- 778 Fischl, B., van der Kouwe, A., Destrieux, C., Halgren, E., Ségonne, F., Salat, D. H., et al. (2004b).
779 Automatically Parcellating the Human Cerebral Cortex. *Cerebral Cortex* 14, 11–22.
780 doi:10.1093/cercor/bhg087.
- 781 Fonov, V., Evans, A. C., Botteron, K., Almli, C. R., McKinstry, R. C., and Collins, D. L. (2011).
782 Unbiased average age-appropriate atlases for pediatric studies. *NeuroImage* 54, 313–327.
783 doi:10.1016/j.neuroimage.2010.07.033.
- 784 Fonov, V., Evans, A., McKinstry, R., Almli, C., and Collins, D. (2009). Unbiased nonlinear average
785 age-appropriate brain templates from birth to adulthood. *NeuroImage* 47, S102. doi:10.1016/S1053-
786 8119(09)70884-5.
- 787 Foulon, C., Cerliani, L., Kinkingnéhun, S., Levy, R., Rosso, C., Urbanski, M., et al. (2018). Advanced
788 lesion symptom mapping analyses and implementation as BCBtoolkit. *GigaScience* 7.
789 doi:10.1093/gigascience/giy004.
- 790 Frackowiak, R. S. J., Friston, K. J., Frith, C., Dolan, R., and Mazziotta, J. C. eds. (1997). *Human Brain*
791 *Function*. Academic Press USA Available at: <http://www.fil.ion.ucl.ac.uk/spm/doc/books/hbf1/>.
- 792 Friston, K., Ashburner, J., Kiebel, S., Nichols, T., and Penny, W. (2007). *Statistical Parametric*
793 *Mapping*. Elsevier doi:10.1016/B978-0-12-372560-8.X5000-1.
- 794 Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., van der Walt, S., Descoteaux, M., et al. (2014).
795 Dipy, a library for the analysis of diffusion MRI data. *Front. Neuroinform.* 8.
796 doi:10.3389/fninf.2014.00008.
- 797 Gorgolewski, K., Burns, C. D., Madison, C., Clark, D., Halchenko, Y. O., Waskom, M. L., et al. (2011).
798 Nipype: A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python.
799 *Front. Neuroinform.* 5. doi:10.3389/fninf.2011.00013.
- 800 Gorgolewski, K. J., Alfaro-Almagro, F., Auer, T., Bellec, P., Capotă, M., Chakravarty, M. M., et al.

- 801 (2017). BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data
802 analysis methods. *PLoS Computational Biology* 13, e1005209. doi:10.1371/journal.pcbi.1005209.
- 803 Gorgolewski, K. J., Auer, T., Calhoun, V. D., Craddock, R. C., Das, S., Duff, E. P., et al. (2016). The
804 brain imaging data structure, a format for organizing and describing outputs of neuroimaging
805 experiments. *Scientific Data* 3, 160044. doi:10.1038/sdata.2016.44.
- 806 Gousias, I. S., Rueckert, D., Heckemann, R. A., Dyet, L. E., Boardman, J. P., Edwards, A. D., et al.
807 (2008). Automatic segmentation of brain MRIs of 2-year-olds into 83 regions of interest. *NeuroImage*
808 40, 672–684. doi:10.1016/j.neuroimage.2007.11.034.
- 809 Halchenko, Y. O., and Hanke, M. (2012). Open is Not Enough. Let's Take the Next Step: An
810 Integrated, Community-Driven Computing Platform for Neuroscience. *Front. Neuroinform.* 6.
811 doi:10.3389/fninf.2012.00022.
- 812 Hammers, A., Allom, R., Koeppe, M. J., Free, S. L., Myers, R., Lemieux, L., et al. (2003). Three-
813 dimensional maximum probability atlas of the human brain, with particular reference to the temporal
814 lobe. *Human Brain Mapping* 19, 224–247. doi:10.1002/hbm.10123.
- 815 Hua, K., Zhang, J., Wakana, S., Jiang, H., Li, X., Reich, D. S., et al. (2008). Tract probability maps in
816 stereotaxic spaces: Analyses of white matter anatomy and tract-specific quantification. *NeuroImage*
817 39, 336–347. doi:10.1016/j.neuroimage.2007.07.053.
- 818 Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W., and Smith, S. M. (2012). FSL.
819 *Neuroimage* 62, 782–790. doi:10.1016/j.neuroimage.2011.09.015.
- 820 Joliot, M., Jobard, G., Naveau, M., Delcroix, N., Petit, L., Zago, L., et al. (2015). AICHA: An atlas of
821 intrinsic connectivity of homotopic areas. *Journal of Neuroscience Methods* 254, 46–59.
822 doi:10.1016/j.jneumeth.2015.07.013.
- 823 Jones, E., Oliphant, T., Peterson, P., and others (2001). *SciPy: Open source scientific tools for Python*.
824 Available at: <http://www.scipy.org/>.
- 825 Keshavan, A., Datta, E., M. McDonough, I., Madan, C. R., Jordan, K., and Henry, R. G. (2018).
826 Mindcontrol: A web application for brain segmentation quality control. *NeuroImage* 170, 365–372.
827 doi:10.1016/j.neuroimage.2017.03.055.
- 828 Leow, A. D., Yanovsky, I., Chiang, M. C., Lee, A. D., Klunder, A. D., Lu, A., et al. (2007). Statistical
829 Properties of Jacobian Maps and the Realization of Unbiased Large-Deformation Nonlinear Image
830 Registration. *IEEE Transactions on Medical Imaging* 26, 822–832. doi:10.1109/TMI.2007.892646.
- 831 Marcoux, A., Burgos, N., Bertrand, A., Teichmann, M., Routier, A., Wen, J., et al. (2018). An
832 Automated Pipeline for the Analysis of PET Data on the Cortical Surface. *Front. Neuroinform.* 12.
833 doi:10.3389/fninf.2018.00094.
- 834 McDermott, M. B. A., Wang, S., Marinsek, N., Ranganath, R., Ghassemi, M., and Foschini, L. (2019).
835 Reproducibility in Machine Learning for Health. in *ICLR 2019 Reproducibility in Machine Learning*
836 *Workshop* Available at: <http://arxiv.org/abs/1907.01463> [Accessed March 26, 2021].
- 837 McKinney, W. (2010). Data Structures for Statistical Computing in Python. in *Proceedings of the 9th*
838 *Python in Science Conference*, eds. S. van der Walt and J. Millman, 51–56.
- 839 Mori, S., Wakana, S., Nagae-Poetscher, L., and van Zijl, P. (2005). *MRI Atlas of Human White Matter*.
840 Amsterdam: Elsevier.
- 841 Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). “PyTorch: An
842 Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information*

- 843 *Processing Systems 32*, eds. H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-
844 Buc, E. Fox, and R. Garnett (Curran Associates, Inc.), 8024–8035. Available at:
845 [http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-](http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf)
846 [library.pdf](http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf).
- 847 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-
848 learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825–2830.
- 849 Poldrack, R. A., Baker, C. I., Durnez, J., Gorgolewski, K. J., Matthews, P. M., Munafò, M. R., et al.
850 (2017). Scanning the horizon: towards transparent and reproducible neuroimaging research. *Nat Rev*
851 *Neurosci* 18, 115–126. doi:10.1038/nrn.2016.167.
- 852 Poline, J.-B., Breeze, J. L., Ghosh, S. S., Gorgolewski, K., Halchenko, Y. O., Hanke, M., et al. (2012).
853 Data sharing in neuroimaging research. *Front. Neuroinform.* 6. doi:10.3389/fninf.2012.00009.
- 854 Reuter, M., Rosas, H. D., and Fischl, B. (2010). Highly accurate inverse consistent registration: A
855 robust approach. *NeuroImage* 53, 1181–1196. doi:10.1016/j.neuroimage.2010.07.020.
- 856 Reuter, M., Schmansky, N. J., Rosas, H. D., and Fischl, B. (2012). Within-subject template estimation
857 for unbiased longitudinal image analysis. *NeuroImage* 61, 1402–1418.
858 doi:10.1016/j.neuroimage.2012.02.084.
- 859 Samper-González, J., Burgos, N., Bottani, S., Fontanella, S., Lu, P., Marcoux, A., et al. (2018).
860 Reproducible evaluation of classification methods in Alzheimer’s disease: Framework and application
861 to MRI and PET data. *NeuroImage* 183, 504–521. doi:10.1016/j.neuroimage.2018.08.042.
- 862 Savio, A. M., Schutte, M., Graña, M., and Yakushev, I. (2017). Pypes: Workflows for Processing
863 Multimodal Neuroimaging Data. *Front Neuroinform* 11. doi:10.3389/fninf.2017.00025.
- 864 Shattuck, D. W., Mirza, M., Adisetiyo, V., Hojatkashani, C., Salamon, G., Narr, K. L., et al. (2008).
865 Construction of a 3D probabilistic atlas of human cortical structures. *NeuroImage* 39, 1064–1080.
866 doi:10.1016/j.neuroimage.2007.09.031.
- 867 Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep Inside Convolutional Networks:
868 Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034 [cs]*. Available at:
869 <http://arxiv.org/abs/1312.6034> [Accessed July 4, 2018].
- 870 Thomas, B. A., Cuplov, V., Bousse, A., Mendes, A., Thielemans, K., Hutton, B. F., et al. (2016).
871 PETPVC: a toolbox for performing partial volume correction techniques in positron emission
872 tomography. *Physics in Medicine and Biology* 61, 7975–7993. doi:10.1088/0031-9155/61/22/7975.
- 873 Tournier, J.-D., Calamante, F., and Connelly, A. (2007). Robust determination of the fibre orientation
874 distribution in diffusion MRI: Non-negativity constrained super-resolved spherical deconvolution.
875 *NeuroImage* 35, 1459–1472. doi:10.1016/j.neuroimage.2007.02.016.
- 876 Tournier, J.-D., Calamante, F., and Connelly, A. (2010). Improved probabilistic streamlines
877 tractography by 2nd order integration over fibre orientation distributions. in *Proceedings of the*
878 *International Society for Magnetic Resonance in Medicine* Available at:
879 https://cds.ismrm.org/protected/10MProceedings/files/1670_4298.pdf [Accessed January 25, 2019].
- 880 Tournier, J.-D., Calamante, F., and Connelly, A. (2012). MRtrix: Diffusion tractography in crossing
881 fiber regions. *Int. J. Imaging Syst. Technol.* 22, 53–66. doi:10.1002/ima.22005.
- 882 Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., et al. (2010).
883 N4ITK: Improved N3 Bias Correction. *IEEE Transactions on Medical Imaging* 29, 1310–1320.
884 doi:10.1109/TMI.2010.2046908.

- 885 Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., et al.
886 (2002). Automated Anatomical Labeling of Activations in SPM Using a Macroscopic Anatomical
887 Parcellation of the MNI MRI Single-Subject Brain. *NeuroImage* 15, 273–289.
888 doi:10.1006/nimg.2001.0978.
- 889 van der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient
890 Numerical Computation. *Comput. Sci. Eng.* 13, 22–30. doi:10.1109/MCSE.2011.37.
- 891 Wakana, S., Caprihan, A., Panzenboeck, M. M., Fallon, J. H., Perry, M., Gollub, R. L., et al. (2007).
892 Reproducibility of quantitative tractography methods applied to cerebral white matter. *NeuroImage* 36,
893 630–644. doi:10.1016/j.neuroimage.2007.02.049.
- 894 Wen, J., Samper-González, J., Bottani, S., Routier, A., Burgos, N., Jacquemont, T., et al. (2021).
895 Reproducible Evaluation of Diffusion MRI Features for Automatic Classification of Patients with
896 Alzheimer’s Disease. *Neuroinformatics* 19, 57–78. doi:10.1007/s12021-020-09469-5.
- 897 Wen, J., Thibeau-Sutre, E., Diaz-Melo, M., Samper-González, J., Routier, A., Bottani, S., et al. (2020).
898 Convolutional Neural Networks for Classification of Alzheimer’s Disease: Overview and
899 Reproducible Evaluation. *Medical Image Analysis*, 101694. doi:10.1016/j.media.2020.101694.
- 900 Worsley, K., Taylor, J., Carbonell, F., Chung, M., Duerden, E., Bernhardt, B., et al. (2009). SurfStat:
901 A Matlab toolbox for the statistical analysis of univariate and multivariate surface and volumetric data
902 using linear mixed effects models and random field theory. *NeuroImage* 47, S102. doi:10.1016/S1053-
903 8119(09)70882-1.
- 904
- 905